Track: CMS Blue Button

INTEROPERABILITY
INSTITUTE

Table of Contents

Introduction to CMS Blue Button

Blue Button allows Medicare beneficiaries to quickly and easily access their medical claims data online. It also allows for the safe and secure exchange of beneficiaries' medical claims between the various applications and programs that they interact with and rely on to help them manage their plan.

Blue Button uses HL7® FHIR® and OAuth 2.0 standards for beneficiary data and identification respectively to ensure that access to and exchange of information is easy and secure. Over 53 Million beneficiaries have at least 1 claim in the Blue Button dataset, including historical claims going back 4 years.
In addition, Blue Button offers a synthetic data set that's provides developers with the chance to test their applications with accurate data without the risk that comes with using the sensitive Private Health Individuals of real beneficiaries.

It contains parts A (inpatient coverage), B (outpatient coverage), and D (prescription drug coverage) data for beneficiaries with traditional Medicare coverage. It also contains Part D data for those on Medicare Advantage plans.

Important Note: For this track each team will be assigned a patient existing in the IOL platform with a unique SSN and collection of information (located in Appendix C of this document). The examples used in this document will not produce the same information as the persona assigned to your team.

Scenario 1- Showcasing Explanation of Benefits (EOBs) and Coverage info from one PIT (SILVERCARE)

Terry needs to see information on his current Medicare coverage. He is unable to find his Medicare card. He would also like to see a list of his EOBs and access information on how to read them.

Action: Explore ways to showcase a patient's medical insurance information, review coverage, and allow him to review their EOBs.

Precondition: Be able to utilize the queries listed below in the Silvercare PIT in IOL.

Success Criteria: Successfully locate a patient's EOB's and coverage information.

Examples- Possible Queries:

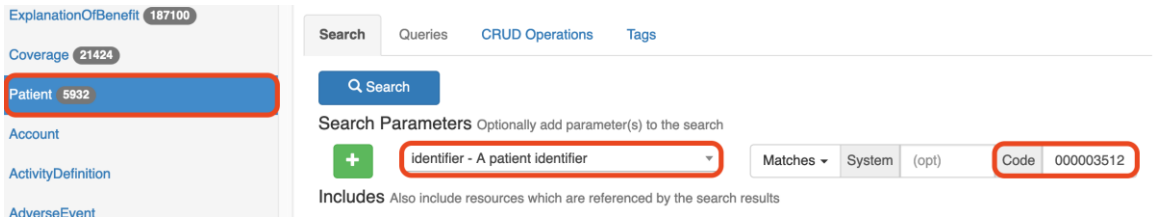
1. Find a patient ID using SSN
2. Locate the patient's EOB's using the *ExplanationOfBenefits* resource.

3. Review the patient's coverage information in the *Coverage* resource

Queries for Scenario 1

Query 1 – Finding A Patient By SSN

Because the HAPI FHIR® Interface only supports a limited set of search parameters, an additional initial query is necessary to find the Patient record by SSN. The id of the Patient record we find will be necessary to execute future queries.



Searching for a patient requires selecting the Patient option from the resource's navigation menu on the left side of the page. Specifying the SSN to search for is then accomplished by selecting the identifier option in the search parameters dropdown, then entering the target SSN in the code field to the right (as seen below SSN 000003512 is used in this example).

Note: In order to perform a bulk query of patient data, simply press "Search" without including the patient identifier in the search parameters.

This query produces the result shown below. Pay particular attention to the lower-right area of the screen, which contains the section labelled Result Body. This is the data returned in response to the query.

The Result Body section in this case contains a line at the top that reads “Bundle contains 1 / 1 entries”. This line shows how many results matched the query that was specified; the first number is the count of results included in this response, the second is the total number of record satisfying that query in this PIT. When a query matches a large number of records, the numbers can be different because the FHIR® server has a limit on the number of records it can return in a single response.

Below the summary section is a section labelled Raw Message. This section contains the actual data sought by the query (in this case, a Patient record identified by the supplied SSN).

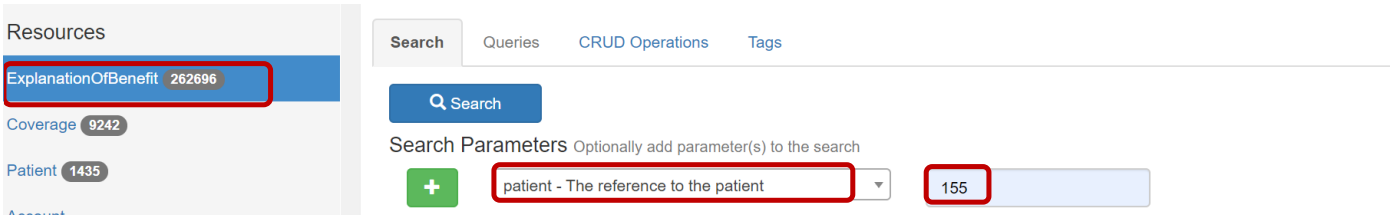
Raw Message


```
{
  "resourceType": "Bundle",
  "id": "5726bf18-5e32-4979-9b2e-3a8a6dd9c699",
  "meta": {
    "lastUpdated": "2020-05-15T15:22:35.307+00:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient?_pretty=true&identifier=000003512"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Patient/155",
      "resource": {
        "resourceType": "Patient",
        "id": "155",
        "meta": {
          "extension": [
            {
              "url": "http://hapifhir.io/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#EbYsLmZVFpgD0k2Q"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:52:25.000+00:00"
        },
        "text": {
          "status": "generated",
          "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\"><div class=\"hapiHeaderText\">Terry Bryan <b>BRUCE </b></div>
<table class=\"hapiPropertyTable\"><tbody><tr><td>Identifier</td><td>000003512</td></tr><tr><td>Date of birth</td><td><span
>01 September 1948</span></td></tr></tbody></table></div>"
        },
        "extension": [
          {
            "url": "http://mihin.org/extension/copyright",
            "valueString": "Copyright 2014-2020 Michigan Health Information Network Shared Services. Licensed under the Apa
che License, Version 2.0 (the 'License'); you may not use this file except in compliance with the License. You may obtain a
copy of the License at  http://www.apache.org/licenses/LICENSE-2.0.  Unless required by applicable law or agreed to in writ
ing, software distributed under the License is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIN
D, either express or implied. See the License for the specific language governing permissions and limitations under the Lic
```


There is a wealth of information about the patient here, but the main field of interest for purposes of this exercise is the id. Specified under the "resource" element of a member of the "entry" collection, this is the unique identifier of this particular patient record within the PIT. It is what other records will use to refer to this patient and thus can be used to search for only records relevant to this patient.

Query 2 – Locating EOB Records for the Patient (Silvercare)

Once Terry's ID is known, it is possible to locate his explanation of benefits records under the *ExplanationOfBenefits* resource. This is accomplished by first selecting the patient option in the search criteria dropdown, then entering the ID from the previous query in the Resource ID field. See below for an example.



This should produce a complete set of Terry's EOB records for this PIT.

Result Body
JSON bundle
(403055 bytes)

Bundle contains 20 / 333 entries Prev Page Next Page

	ID	Updated
Read Update	ExplanationOfBenefit/1458/_history/1	2020-04-17 20:59:07
Read Update	ExplanationOfBenefit/1553/_history/1	2020-04-17 21:05:58
Read Update	ExplanationOfBenefit/1596/_history/1	2020-04-17 21:06:58
Read Update	ExplanationOfBenefit/2200/_history/1	2020-04-17 21:20:53
Read Update	ExplanationOfBenefit/2616/_history/1	2020-04-17 21:28:38
Read Update	ExplanationOfBenefit/2617/_history/1	2020-04-17 21:28:38
Read Update	ExplanationOfBenefit/2847/_history/1	2020-04-17 21:32:43
Read Update	ExplanationOfBenefit/3105/_history/1	2020-04-17 21:36:50
Read Update	ExplanationOfBenefit/3526/_history/1	2020-04-17 21:42:42
Read Update	ExplanationOfBenefit/4005/_history/1	2020-04-17 21:49:14
Read Update	ExplanationOfBenefit/4358/_history/1	2020-04-17 21:53:38
Read Update	ExplanationOfBenefit/4763/_history/1	2020-04-17 21:58:51

Raw Message

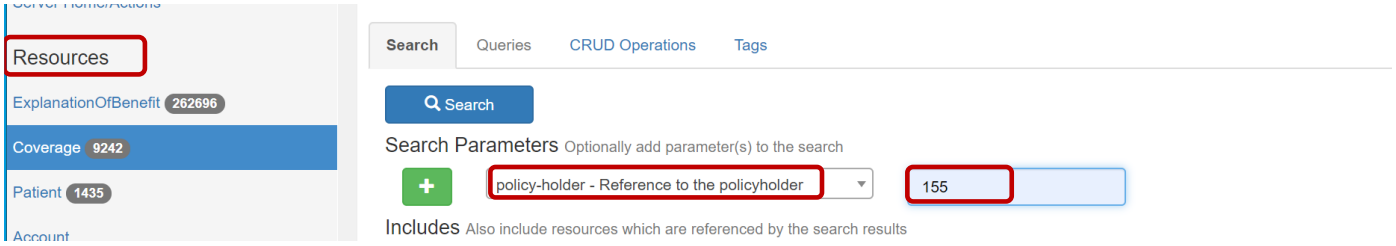

```
{
  "resourceType": "Bundle",
  "id": "e6327297-5bd7-4e93-a276-fcb8570b17f0",
  "meta": {
    "lastUpdated": "2020-05-15T15:24:32.459+00:00"
  },
  "type": "searchset",
  "total": 333,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/ExplanationOfBenefit?_pretty=true&patient=155"
    },
    {
      "relation": "next",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir?_getpages=e6327297-5bd7-4e93-a276-fcb8570b17f0&_getp
offset=20&_count=20&_pretty=true&_bundletype=searchset"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/ExplanationOfBenefit/1458",
      "resource": {
        "resourceType": "ExplanationOfBenefit",
        "id": "1458",
        "meta": {
          "extension": [
            {
              "url": "http://hapifhir.io/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#kU7KJCKB7xr6ZuT4"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:59:07.000+00:00"
        },
        "extension": [
```


MiHIN MICHIGAN HEALTH INFORMATION NETWORK SHARED SERVICES

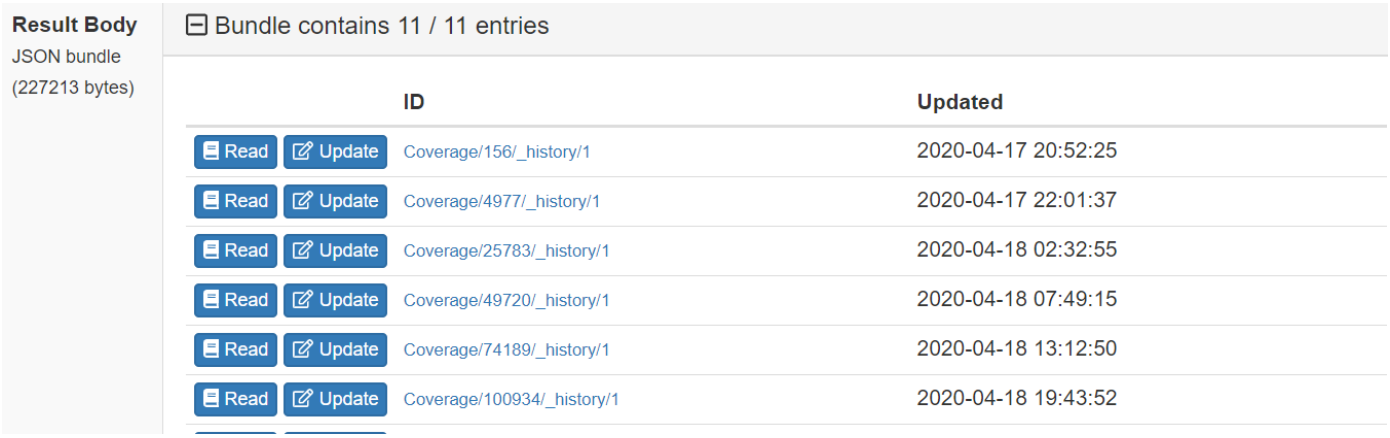
INTEROPERABILITY INSTITUTE

Query 3 – Finding Coverage information for the Patient

Coverage information can be found in a very similar method by using the *beneficiary* option with the ID we used for Query 2.



This should produce Terry's coverage information for this particular PIT.



Raw Message


```
{
  "resourceType": "Bundle",
  "id": "96aba538-2d78-4b56-8add-ac44068bf689",
  "meta": {
    "lastUpdated": "2020-05-15T15:25:56.831+00:00"
  },
  "type": "searchset",
  "total": 11,
  "link": [
    {
      "relation": "self",
      "url": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Coverage?_pretty=true&policy-holder=155"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev-jtx37.devinteropland.com/silvercare/fhir/Coverage/156",
      "resource": {
        "resourceType": "Coverage",
        "id": "156",
        "meta": {
          "extension": [
            {
              "url": "http://hapifhir.io/fhir/StructureDefinition/resource-meta-source",
              "valueUri": "#5TGdZNLlYpUPqqpL"
            }
          ],
          "versionId": "1",
          "lastUpdated": "2020-04-17T20:52:25.000+00:00"
        },
        "extension": [
          {
            "url": "http://mihin.org/extension/copyright",
            "valueString": "Copyright 2014-2020 Michigan Health Information Network Shared Services. Licensed under
che License, Version 2.0 (the 'License'); you may not use this file except in compliance with the License. You may
copy of the License at  http://www.apache.org/licenses/LICENSE-2.0.  Unless required by applicable law or agreed to
ing, software distributed under the License is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
D, either express or implied. See the License for the specific language governing permissions and limitations under
ense."
          },
          {
            "url": "http://hl7.org/fhir/StructureDefinition/coverage-rxbin",
            "valueString": "612030"
          },
```


Scenario 2: Navigating Co-insurance

Terry has secondary insurance to complement his Medicare. How can we make it easier to see what each insurance is paying, and how much Terry is responsible for?

Action: Explore ways to showcase a patient's medical insurance information, review coverage, and allow them to review their EOBs for their secondary insurance.

Precondition: Be able to utilize the queries listed below in the Better Health Insurance PIT in IOL.

Success Criteria: Successfully locate and identify the patient's Medical Insurance and EOB's.

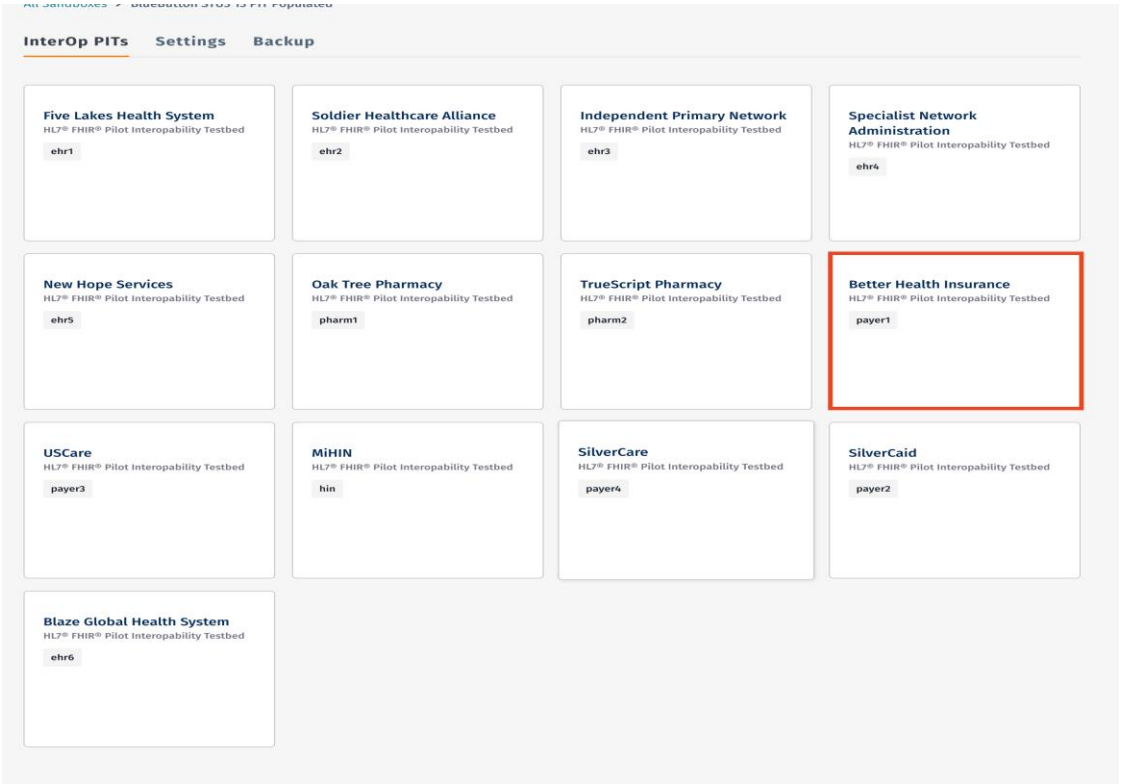
Examples- Possible Queries:

1. Find a patient ID using SSN
2. Locate the patient's EOB's using the *ExplanationOfBenefits* resource.
3. Review the patient's coverage information in the *Coverage* resource

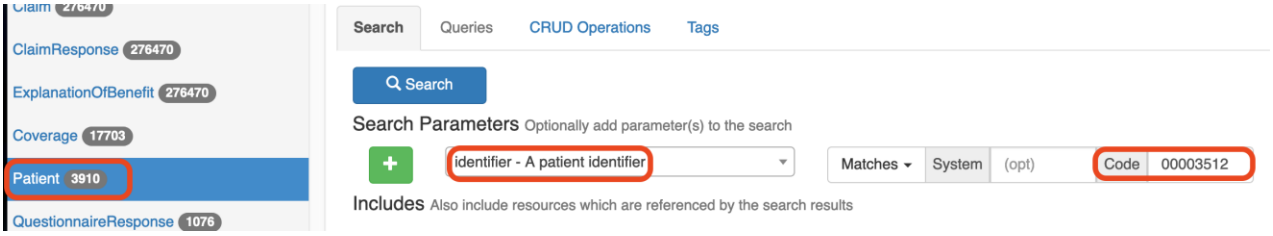
The steps for this scenario are very similar to those for Scenario 1. First, we retrieve the patient's ID using his SSN, then we locate and explore his EOB and Coverage records. The primary difference for this scenario is that we will be performing the search on two payer PIT's, and using the resulting data to build a more complete picture of Terry's insurance profile.

Query 1 – Finding A Patient By SSN

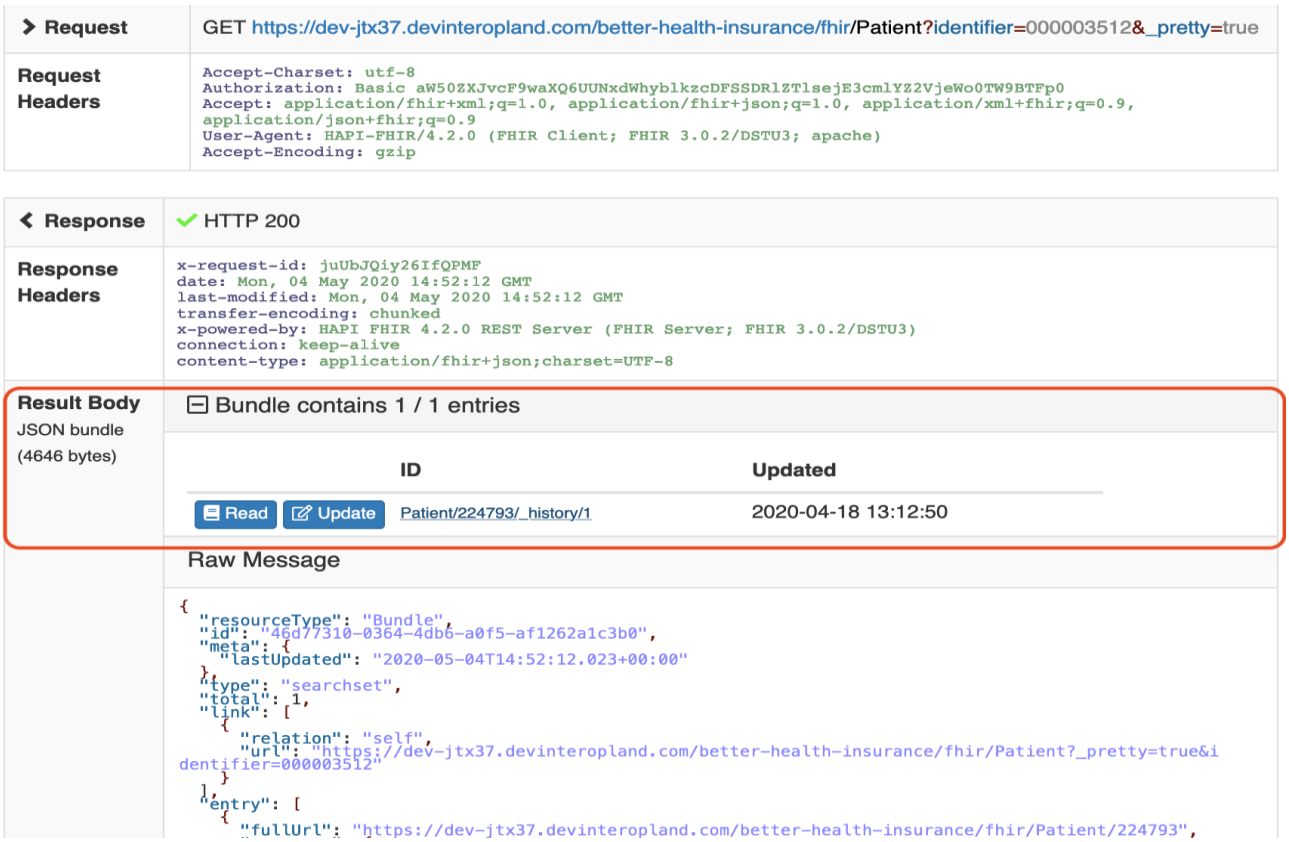
As in Scenario 1, we begin by locating Terry's ID on the payer PIT. This time in addition to the *Silvercare* PIT, we will be searching *Better Health Insurance* for Terry's information. This can be accomplished by first navigating to the Better Health insurance PIT from the sandbox dashboard.



We use the same social security number (000003512) with in the *code* field of the *identifier* option, as pictured below.



We can then make a note of Terry's ID on this second payer PIT. As you can see below, this ID is different from the one in the *Silvercare* PIT, and indeed each PIT has a unique ID for each patient record.



To navigate the Explanation of Benefits resource and review the patient's coverage information, simply repeat the steps for Queries 2 and 3 from Scenario 1 in the Better Health Insurance PIT with the appropriate patient ID for Better Health Insurance.

Scenario 3: Comparing Medication Prices

Action: Pull active meds for patients with Medicaid from Pharmacy PIT and coverage from payer PIT, then cross reference with local pharmacies to identity the best price.

Precondition: Be able to utilize the queries listed below and access third-party prescription services.

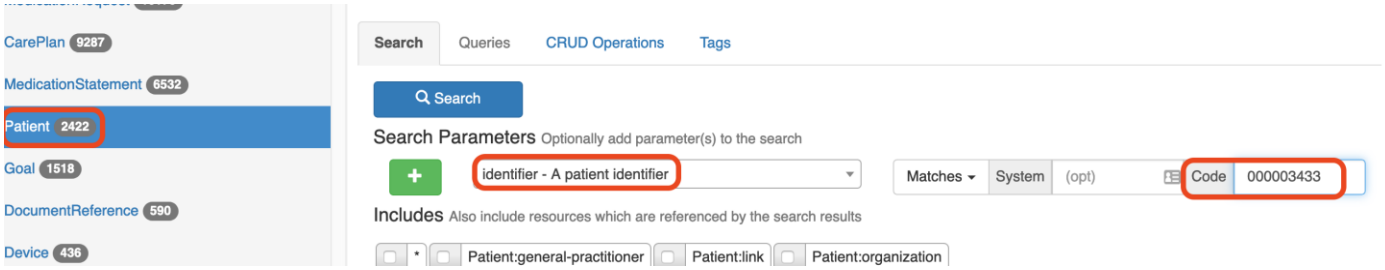
Success Criteria: Successfully compare prices of an active medication for one of the patients in the PIT.

Example: Possible Queries:

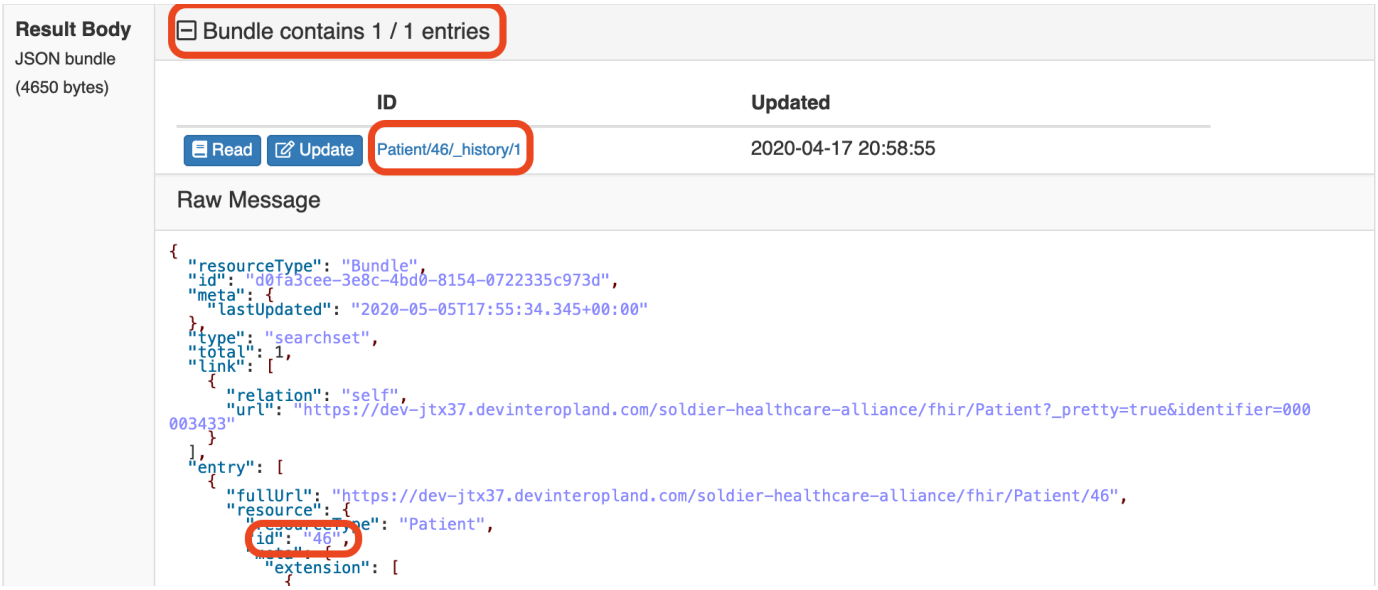
1. Finding the Patient's ID on an EHR PIT
2. Finding Medication Requests for a patient
3. Finding a Patient by SSN in the Silvercare PIT
4. Finding a Patient's Prescriptions in Their EOB Records

Query 1 – Finding the Patient's ID on an EHR PIT (Soldier Healthcare Alliance)

It is also possible to search for medication requests on an EHR PIT. First, like above, we must find the patient's ID on that PIT using their SSN and the *identifier* search option.



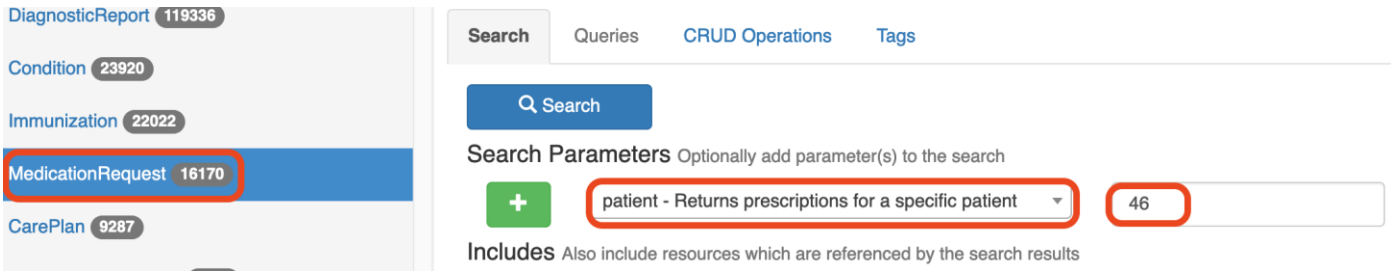
This search returns one result, and provides us with the patient's ID on this PIT.



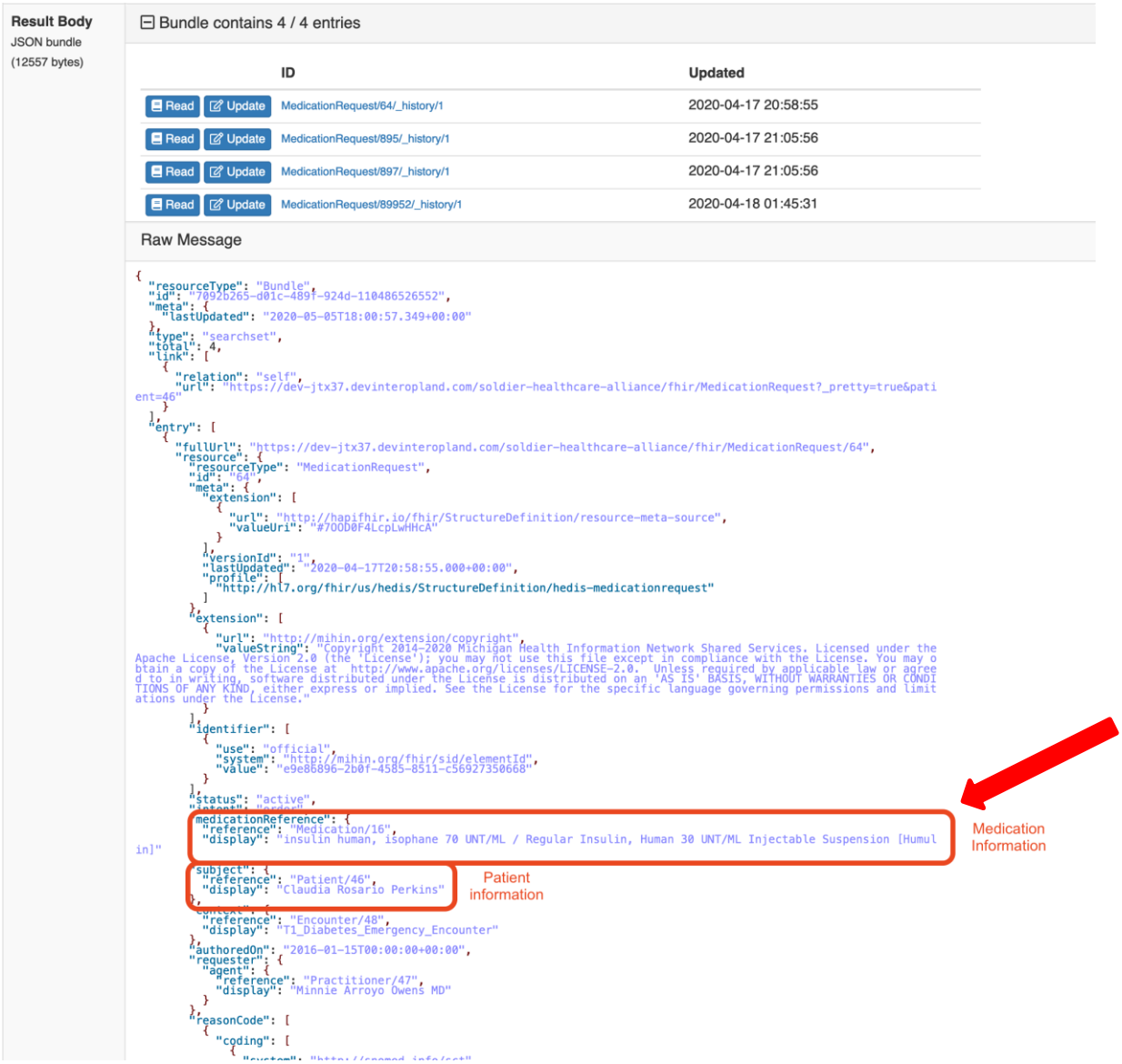
This Identifier will be used in Query 2 to help locate medication requests.

Query 2 – Finding Medication Requests for the patient (Soldier Healthcare Alliance)

A medication request is initiated when a physician or provider orders a new medication. Finding medication requests for the patient can be done by navigating to the *MedicationRequest* resource and selecting the *patient* option. Use the ID acquired in Query 1 and click "Search."



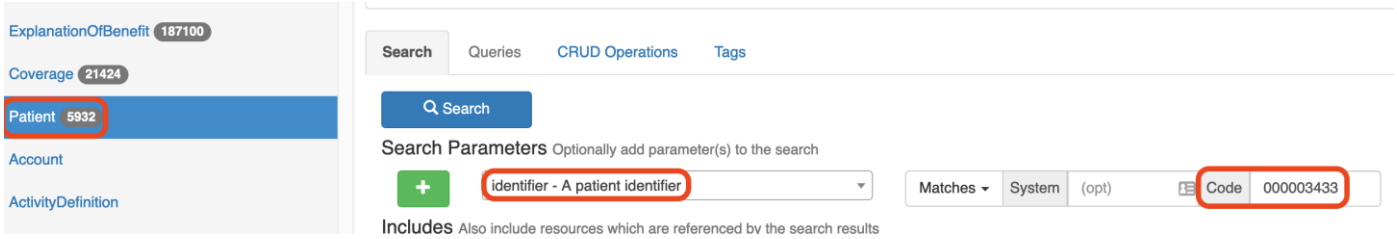
This returns the medication information, which we can then be cross-reference with a third-party service such as GoodRx to provide useful price comparisons.

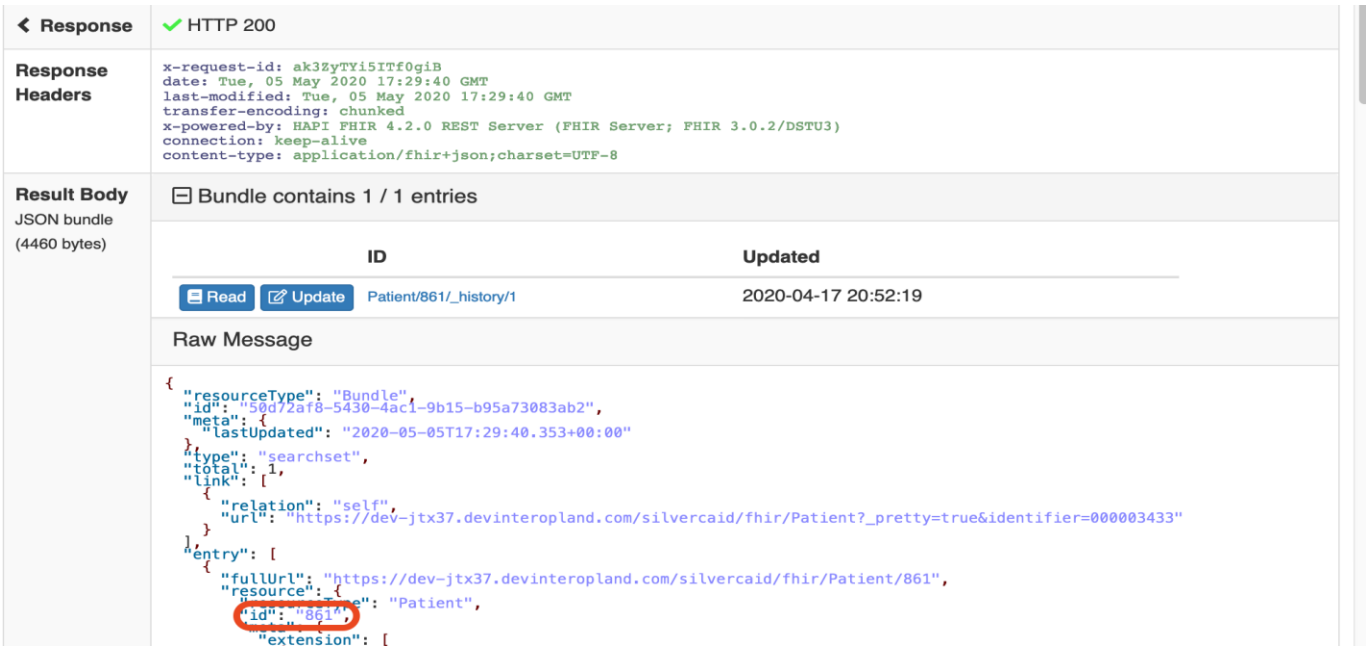


Query 3 -Finding a Patient by SSN in the Silvercare and EHR PIT

Once again, we must locate the patient's ID on the payer PIT. This time in addition to the *Silvercare* PIT, we will also be searching an EHR PIT (Soldier Healthcare Alliance) for *MedicationRequest* records, which we can then use a third-party site to attempt to compare prices.

We use the patient's social security number (000003433) in the *code* field of the *identifier* option, as pictured below.

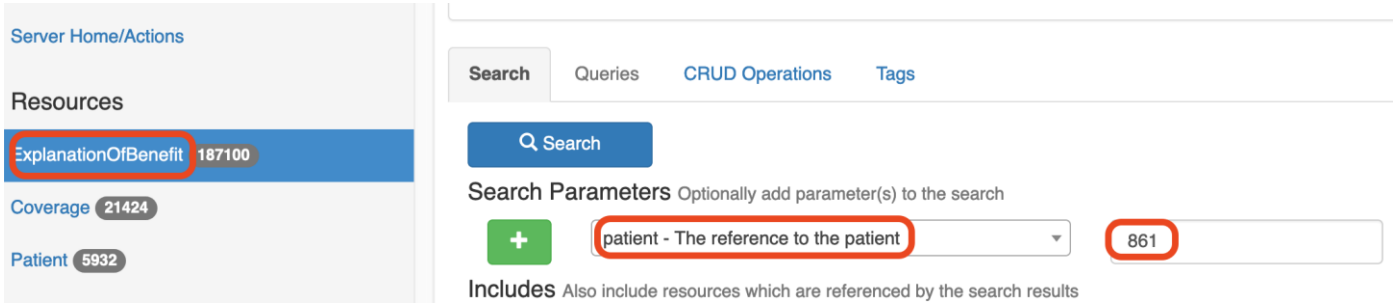




We can then make a note of the ID for use in exploring EOB records for this patient

Query 4- Finding a Patient's Prescriptions in Their EOB Records

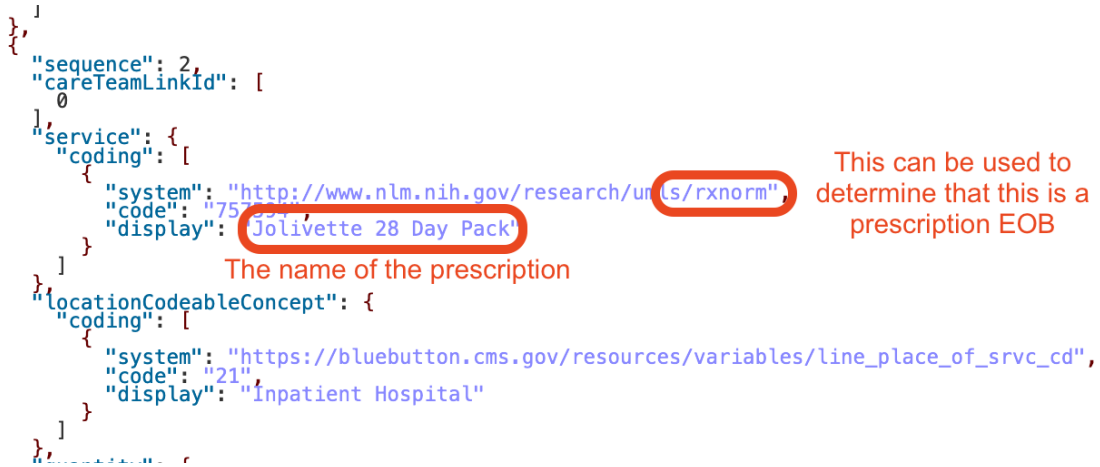
By using the ID gained in the previous query, it is possible to search the patient's EOB records for prescriptions. Select the *patient* option, and enter the patient ID as seen below.



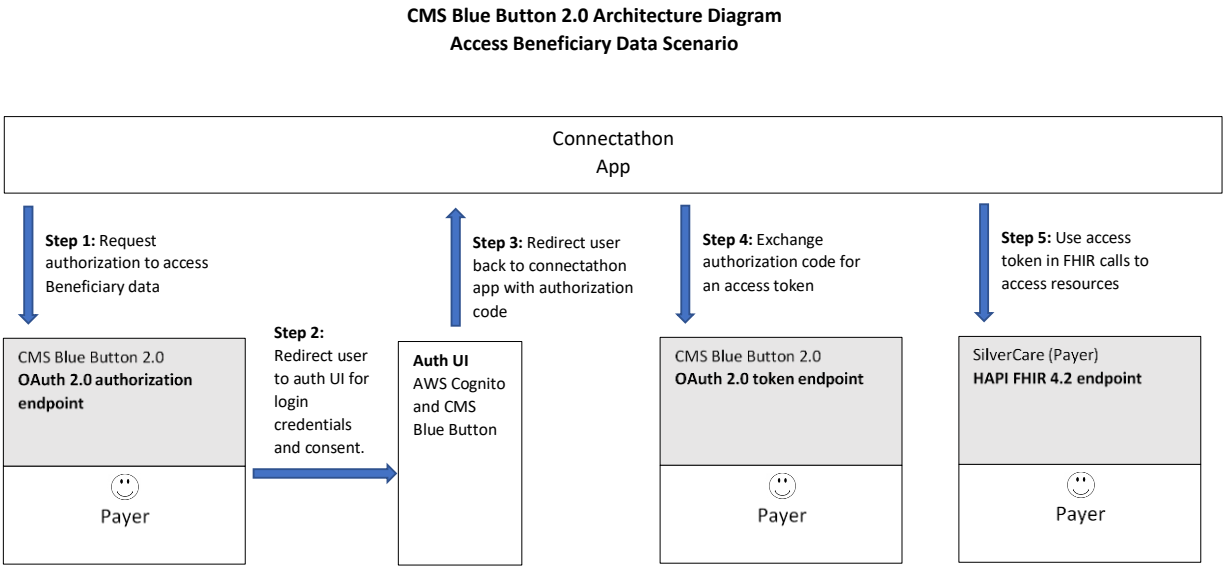
MiHIN MICHIGAN HEALTH INFORMATION NETWORK SHARED SERVICES

INTEROPERABILITY INSTITUTE

Identifying prescription-based EOB's can be done by searching for "**rxnorm**" in the coding section of the record



Appendix A: Steps for Authorization

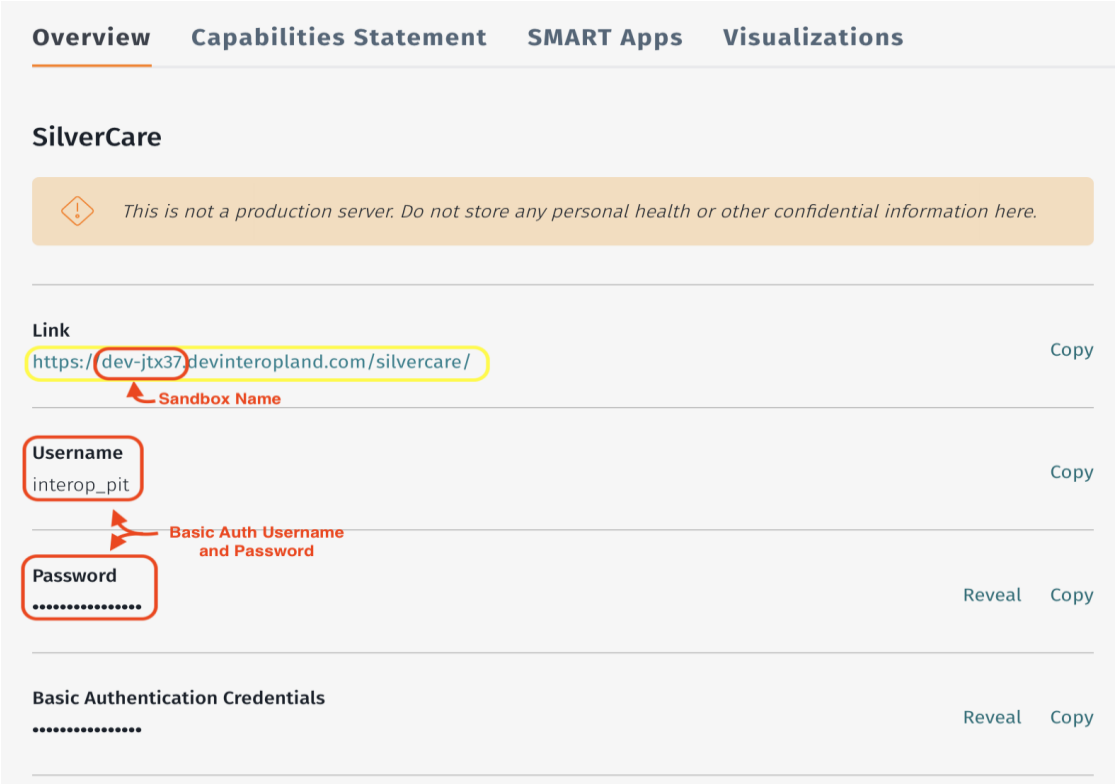


Authorization: In order to query a patient's data as described in the included scenarios, we must complete the process of creating a Blue Button 2.0 application, and authorizing it to access of our sample Patient's data.

Note: For this setup, you will need the username and password of the patient that have been provided for your team.

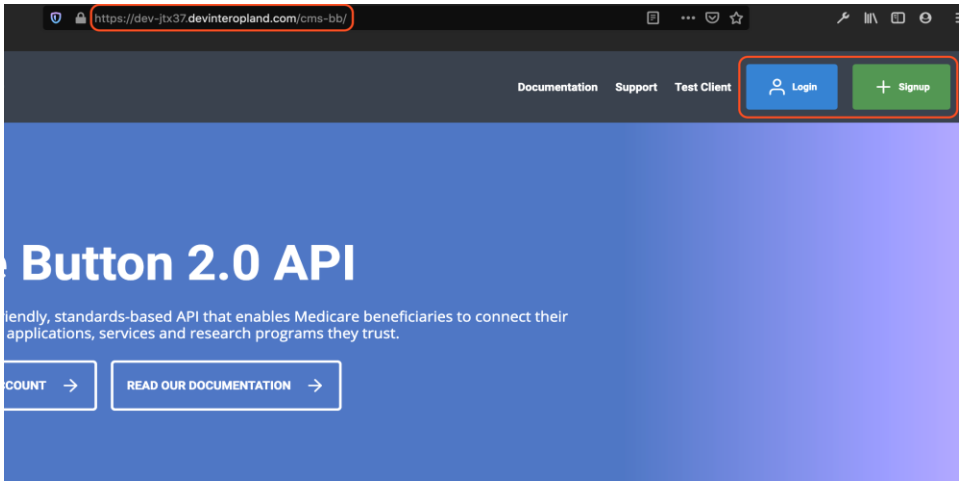
Step 1 - Acquiring the IOL Sandbox Name and URL

First, log into IOL and navigate to the relevant sandbox. Select the Silvercare PIT from the dashboard, and you will be presented with an Overview tab similar to the one below. Note the name and URL of the sandbox for later use. The Basic Auth username and password can be used for bulk querying patient data if needed.

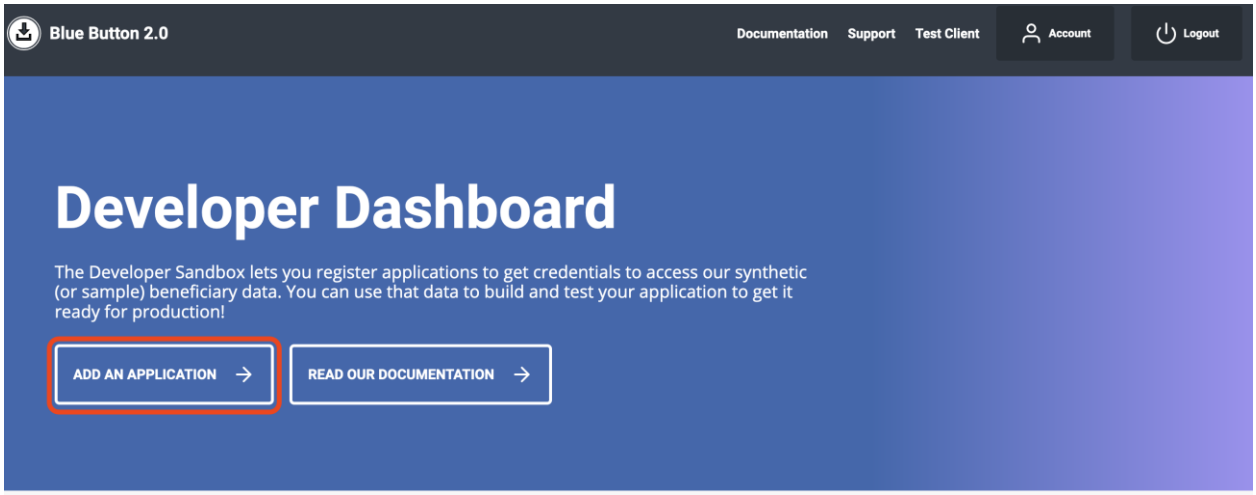


Step 2 - Creating a Blue Button Application

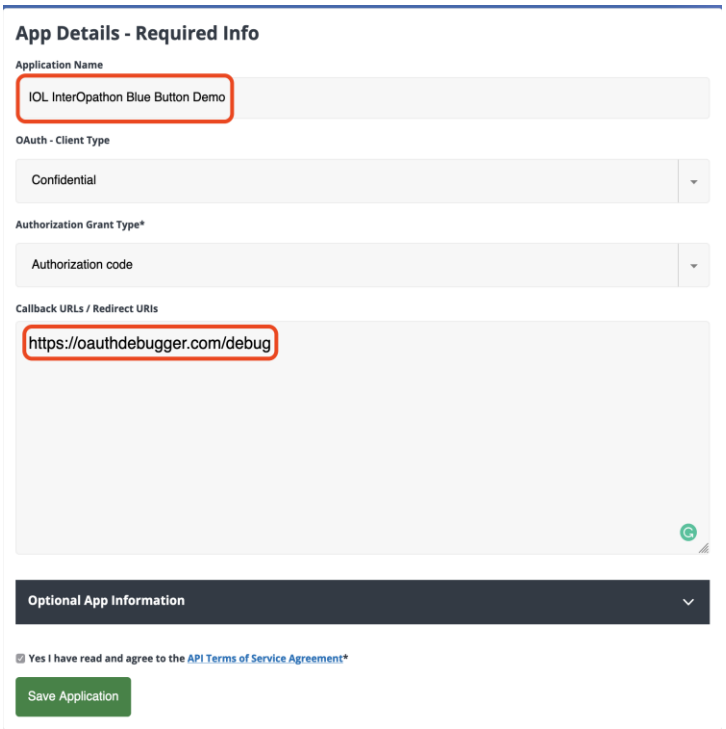
Now that we have the base URL for our sandbox, we can access the Blue Button 2.0 dashboard. Navigate to the address below (replace the sandbox name and domain with the values you acquired in step one). You should see the Blue Button 2.0 dashboard.



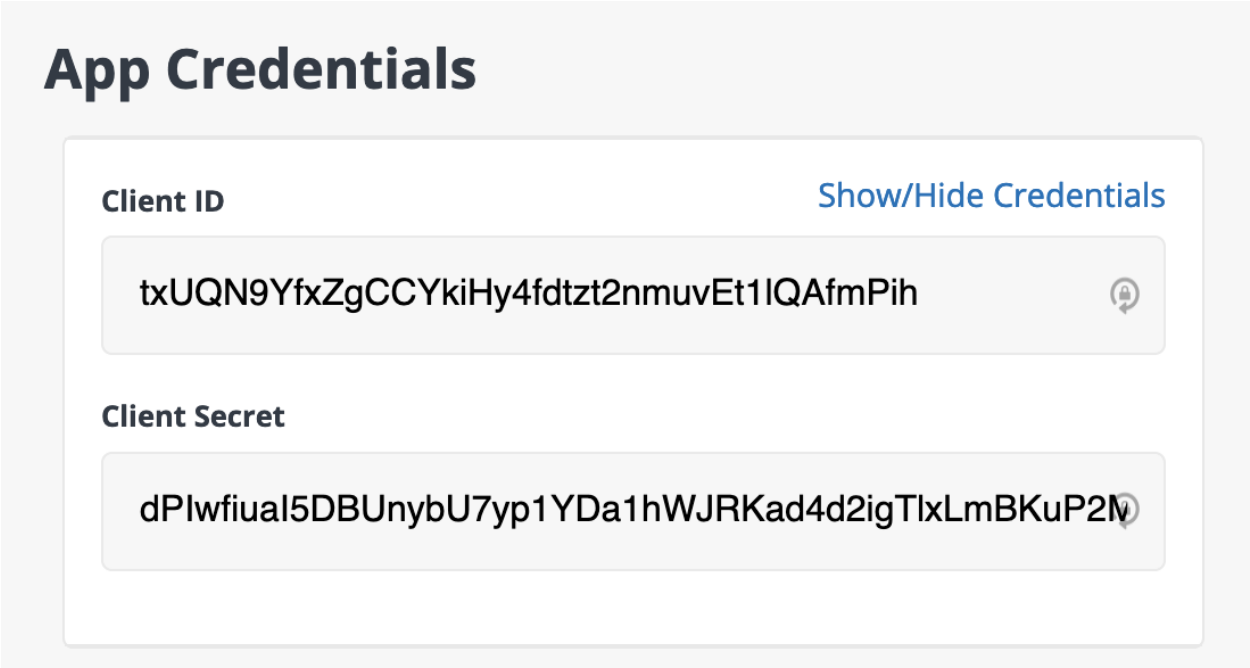
Next we will need to log in or create a new account. Select either option and fill in the form with your information. After you have submitted the form, you will arrive at the Developer Dashboard, where it's possible to create a new Blue Button application.



Click "Add an Application" and fill out the form as shown below. Choose any name for the application, and set the **Callback URLs / Redirect URIs** field to **https://oauthdebugger.com/debug**. This will allow us to inspect the responses to our authorization calls, and obtain a necessary code for use later steps.

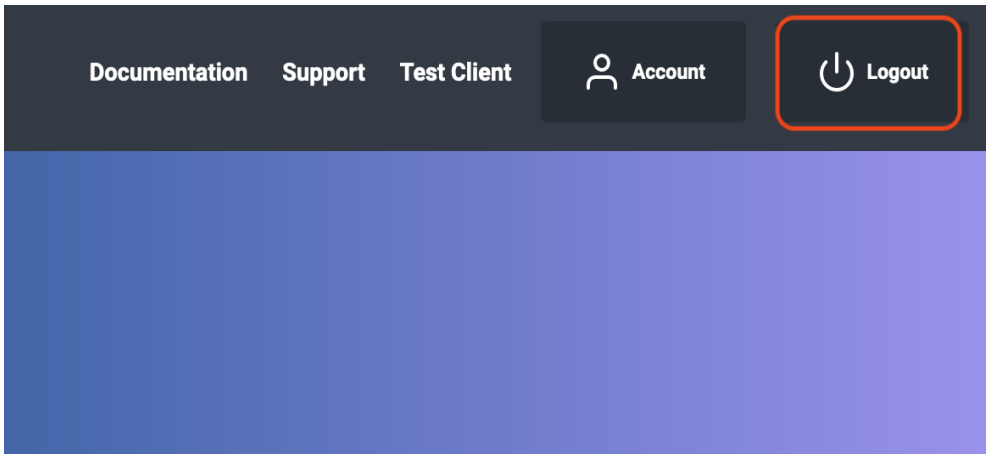


After creating the application, you will be directed to the application details page. The **App Credentials** section should feature a **Client ID** field, as well as a **Client Secret** field. Make a note of both values, as they will be crucial in later steps. For convenience, clicking each field will copy its value to your clipboard.

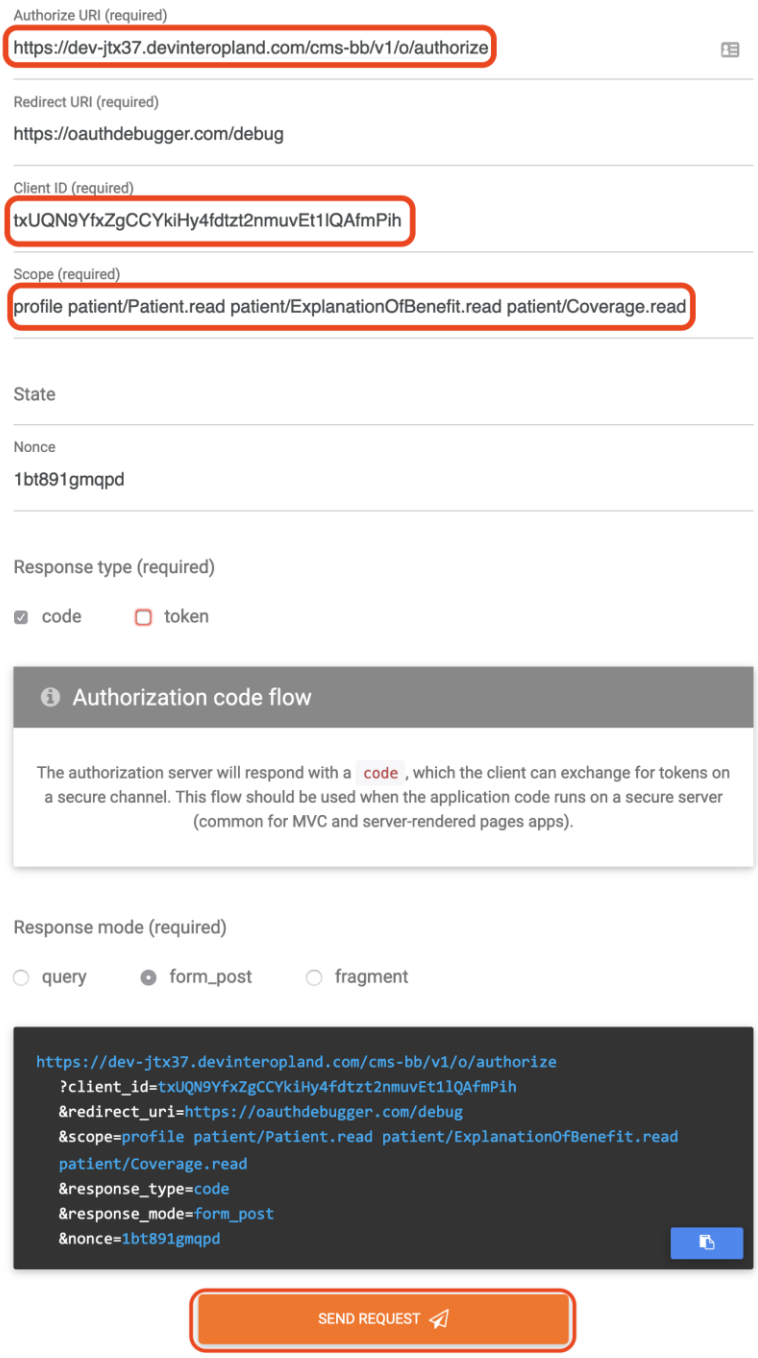


Step 3 - Acquiring the Authorization Code

Now that we have a **Client ID** and **Client Secret** for our application, we need to log in as our sample patient and authorize the application to access the **Patient**, **ExplanationOfBenefits**, and **Coverage** records associated with that patient's account. First, log out of the Blue Button dashboard by pressing "Logout" in the top-right corner of the screen.



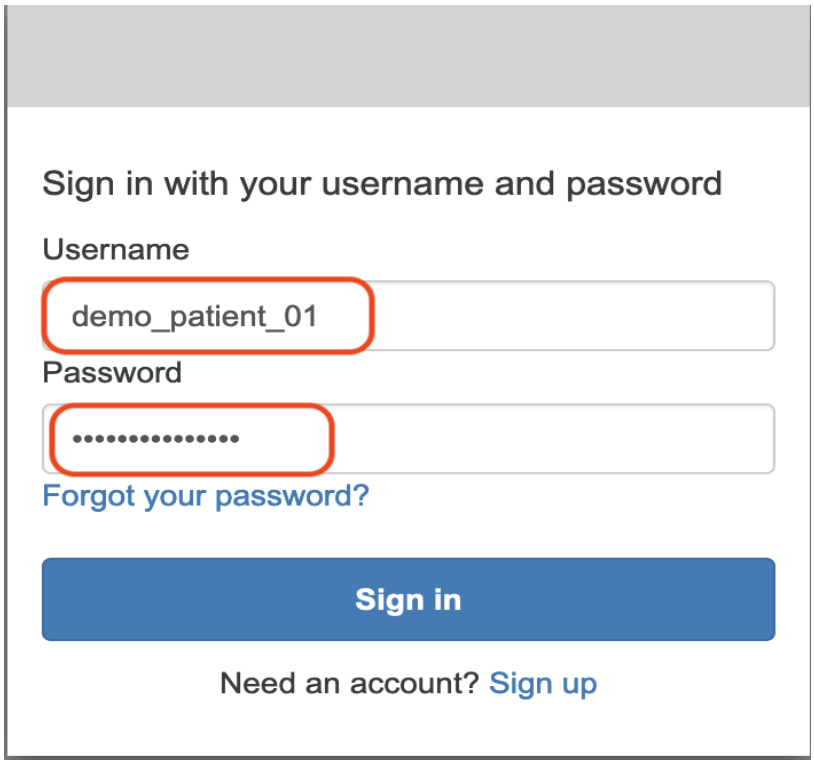
Next, navigate to oauthdebugger.com, where we will configure and run our authorization request. Fill out the form as shown below and press “Send Request”.



Note: For your convenience, the value of the **Scope** field is:

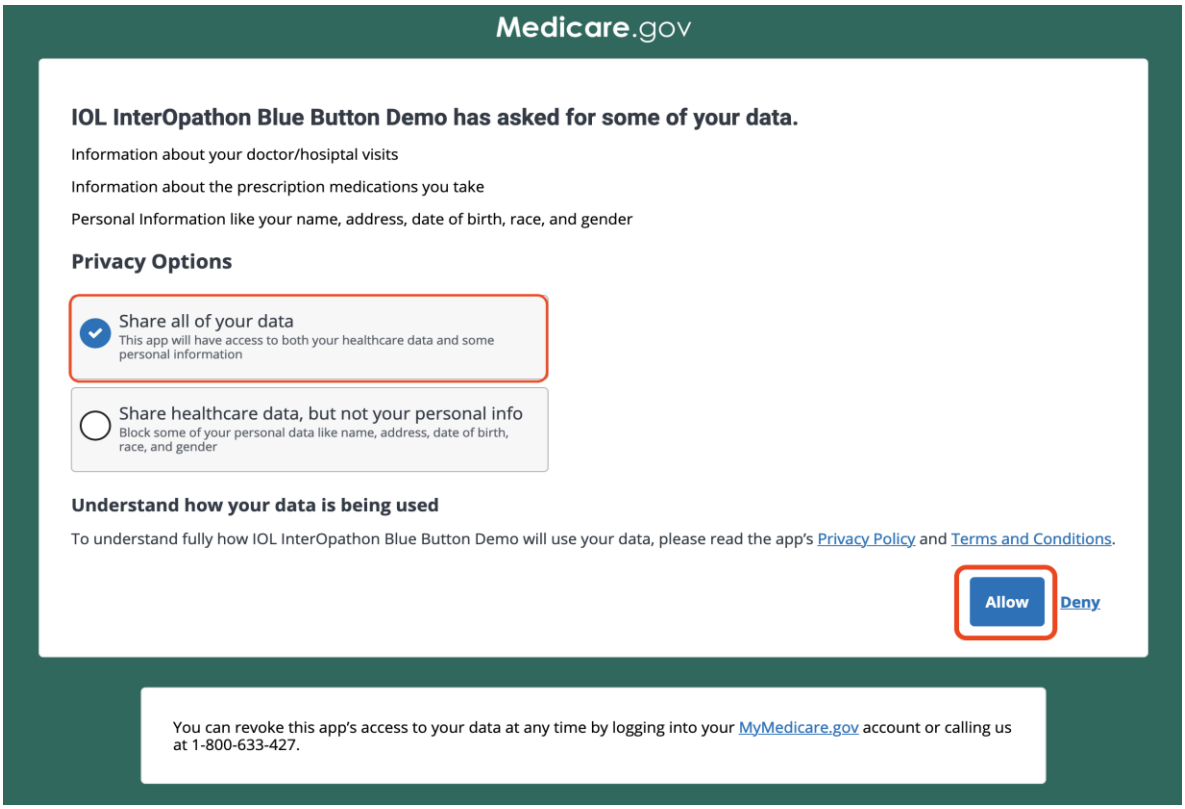
```
profile patient/Patient.read patient/ExplanationOfBenefit.read patient/Coverage.read
```

After sending the request, you will be prompted for a username and password. Since our goal is to authorize our new application to access our sample patient's records, we will use the provided username and password to log in as that patient. Enter the credentials and click "Sign in" to proceed.

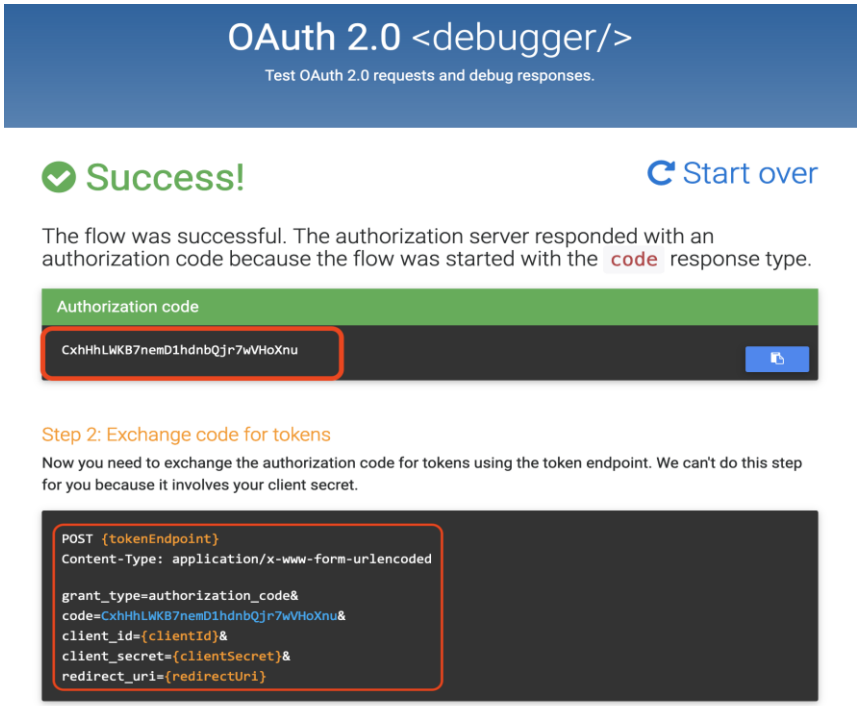


After submitting the patient's credentials, you will be prompted to grant data access permissions. Select "Share all of your data" and click "Allow".

After granting access, you will be shown a response page, which should feature an **Authorization code** as seen in the image below.

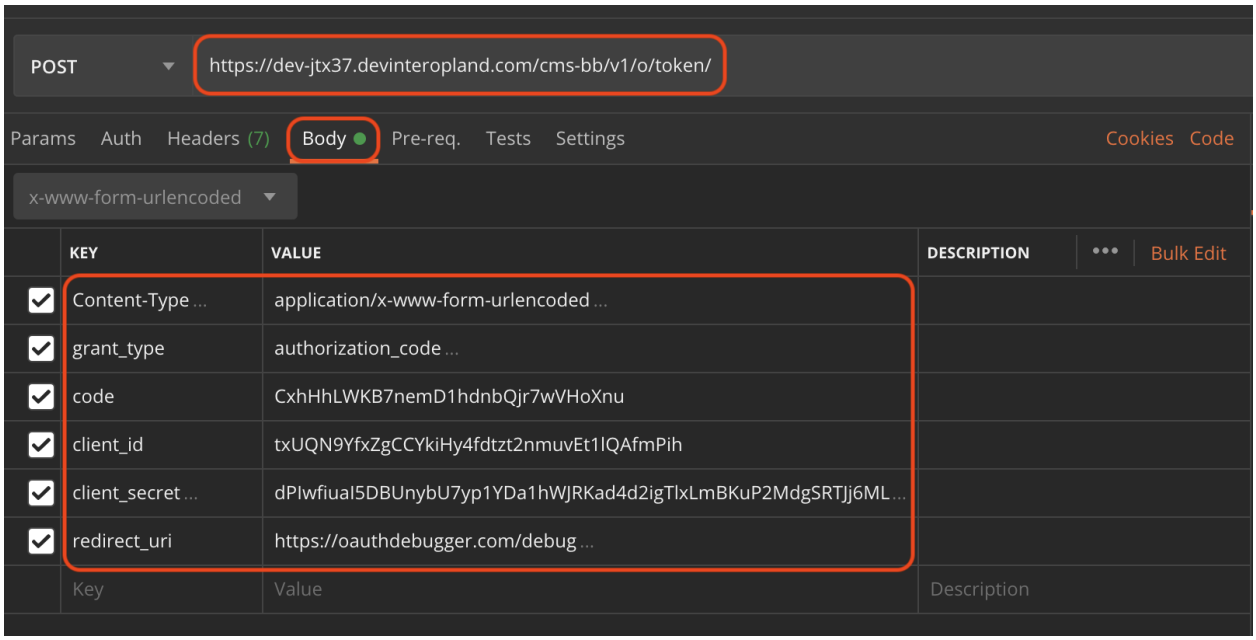


We're also provided with a hint as to our next step. We will need to exchange this code for a bearer token for use in our patient record queries. Make note of the fields listed in the **Step 2** block. They will need to be correctly populated in order to acquire the token.

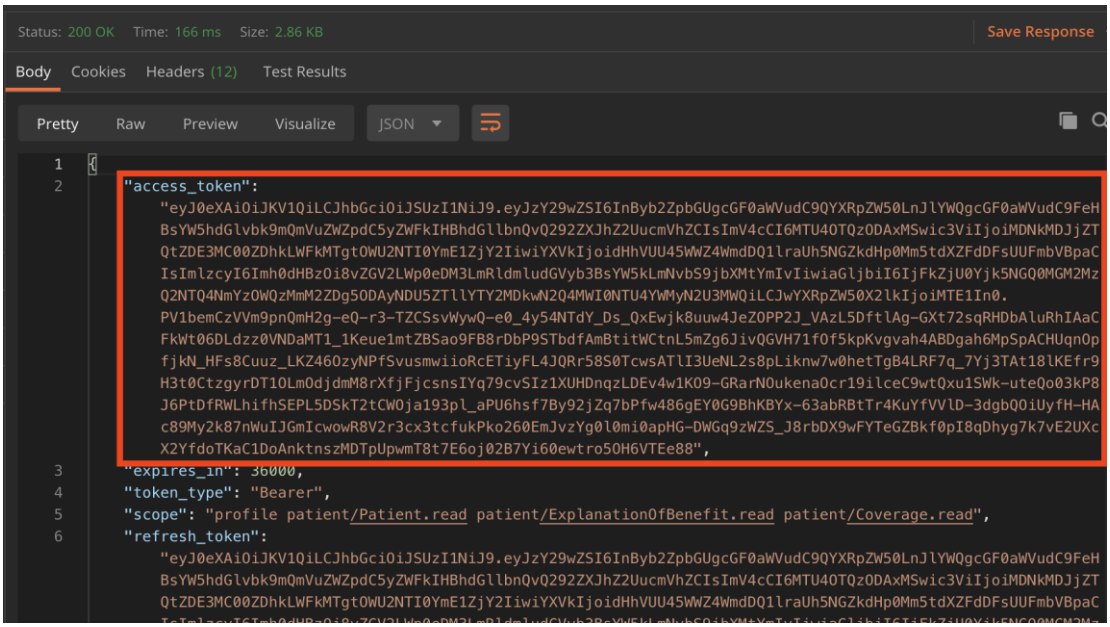


Step 4 - Exchanging the Authorization Code for a Bearer Token

By this point we have collected everything necessary for the token request. Open your API client of choice (we use Postman in this example) and prepare the request as demonstrated below. Substitute the address, code, client_id and client_secret with the values obtained in the previous step.

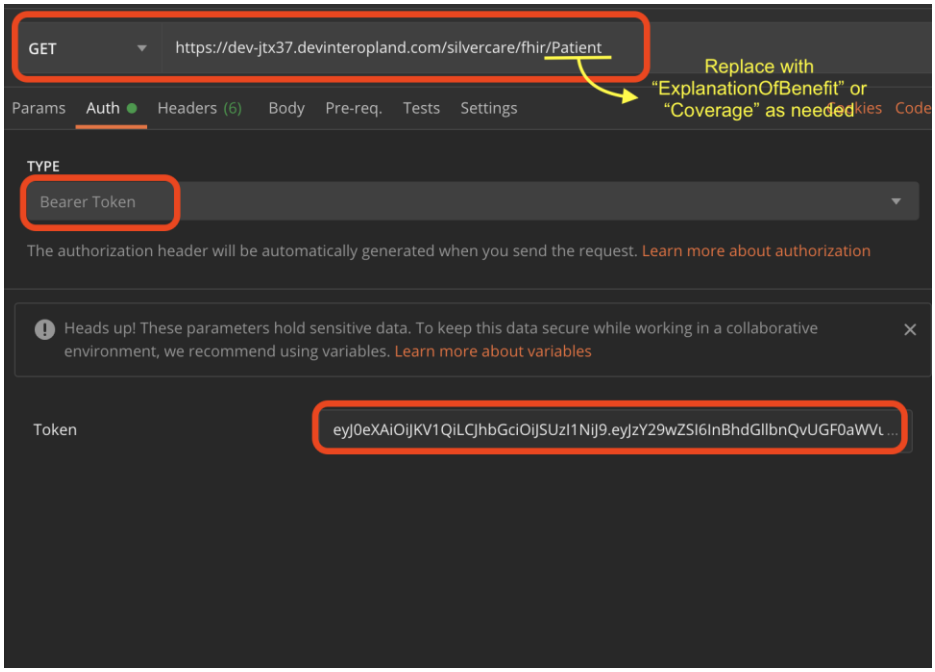


Send the request. The response body should be similar to the following:



The most important value in this response is the **access_token**, which will be passed in the Authentication header with all subsequent requests. Finally, we're ready to access patient records.

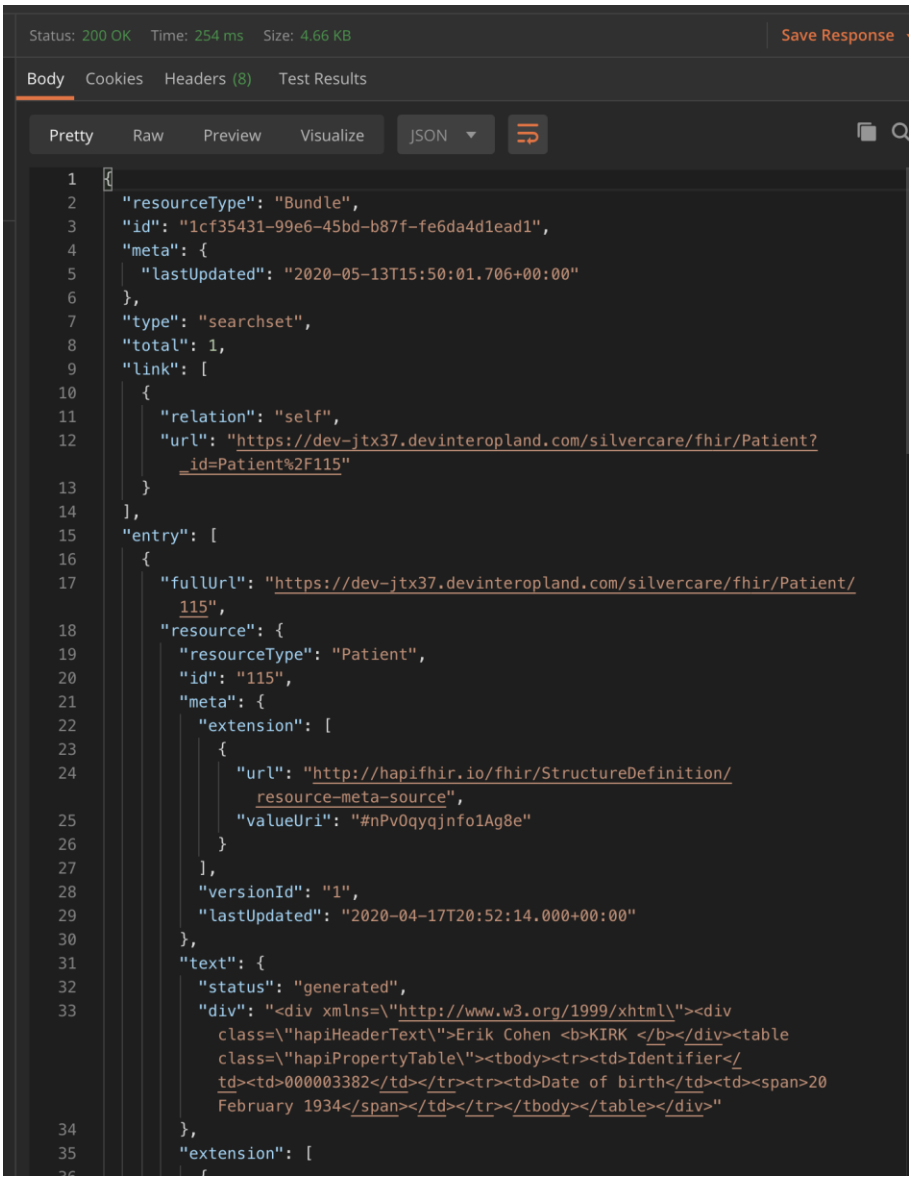
Create a new **GET** request, and set the Auth type to **Bearer Token**. Paste the **access_token** returned by the token endpoint, and send the request.



MiHIN MICHIGAN HEALTH INFORMATION NETWORK SHARED SERVICES

INTEROPERABILITY INSTITUTE

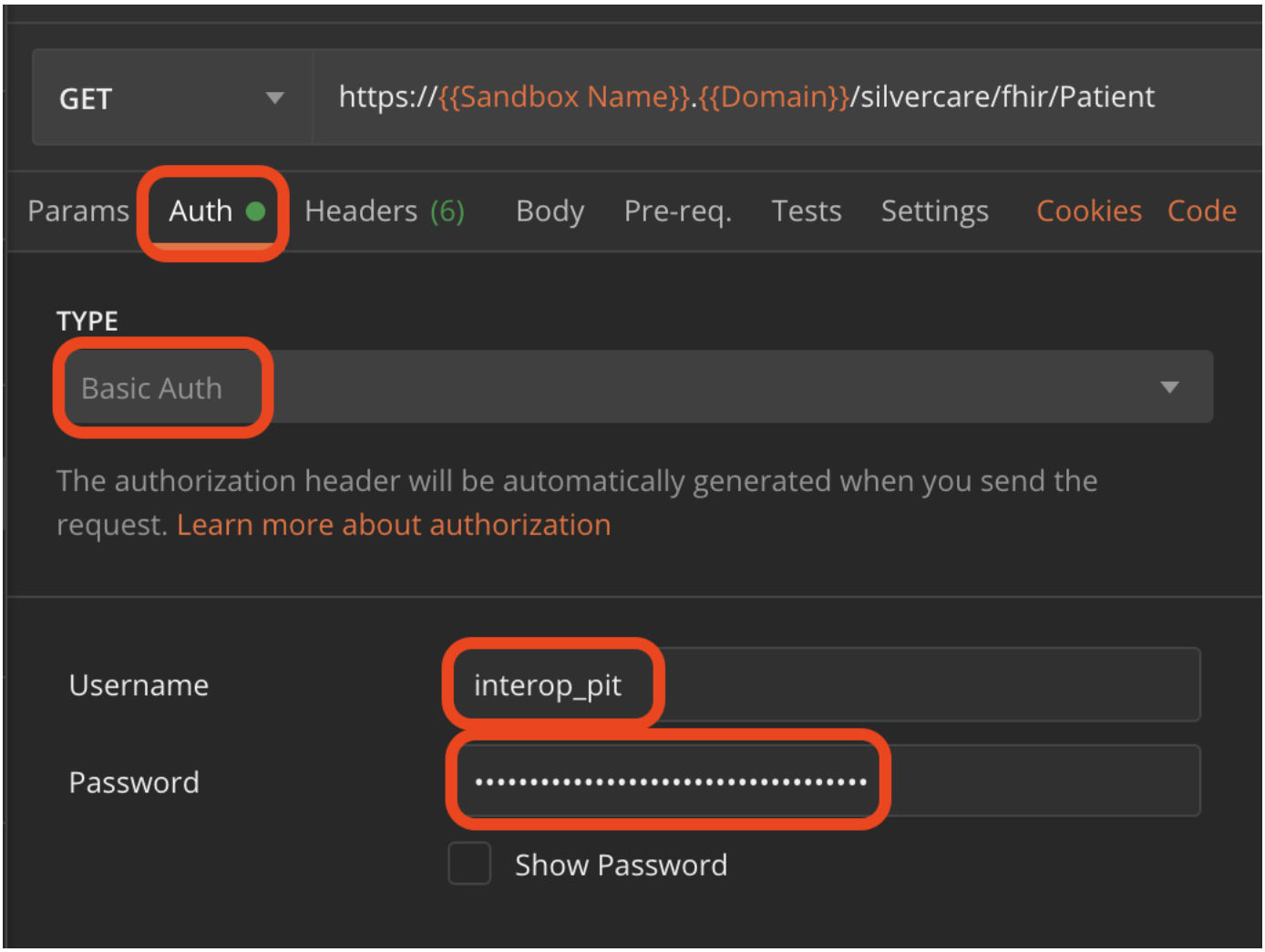
If successful, we should receive the appropriate data block for our sample patient, as shown in the following image.



Note: Replace **Patient** with **ExplanationOfBenefit** or **Coverage** in the URL to facilitate completion of the Blue Button 2.0 scenarios.

Appendix B: Bulk Querying Patient Data

In order to query records for multiple users, you can use the Basic Auth username and password obtained in step 1.



As you can see, our results include all Medicare patients. We can use the same authentication for ExplanationOfBenefits and Coverage records, and we will similarly see unfiltered records for those resources.

Appendix C - Sample Users

ID	SSN	Name	Username	Password
1	000003009	Georgia Burgess	gburgess	!Iolconnect01
15	000003056	Tamara Murillo	tmurillo	!Iolconnect02
25	000003087	Dwight Frey	dfrey	!Iolconnect03
27	000003097	Jennie Moon	jmoon	!Iolconnect04
29	000003111	Nicholas Butler	nbutler	!Iolconnect05
31	000003112	Alicia Atkins	aatkins	!Iolconnect06
33	000003123	Franklin Marks	fmarks	!Iolconnect07
51	000003188	Perry Cortez	pcortez	!Iolconnect08
53	000003189	Jeffrey Gregory	jgregory	!Iolconnect09
55	000003193	Fernando Grant	fgrant	!Iolconnect10
71	000003227	Leah Carson	lcarson	!Iolconnect11
73	000003262	Keith Mooney	kmooney	!Iolconnect12
75	000003265	Benjamin Dixon	bdixon	!Iolconnect13
79	000003303	Mabel Avery	mavery	!Iolconnect14
99	000003357	Dustin Neal	dneal	!Iolconnect15

Judging Criteria

IGNITE
Interoperability: APIs and FHIR® Heat Up

INTEROPERABILITY INSTITUTE

MiHIN MICHIGAN HEALTH INFORMATION NETWORK SHARED SERVICES

Alignment with Track	Helps to improve Inter-operability	Innovation & Creativity	Use of APIs	User Experience	Technical Difficulty	Presentation or Demo
25%	25%	15%	10%	10%	10%	5%
How aligned was the solution with one of the event Tracks?	Does the team clearly show how their solution could be used to improve interoperabil-ity?	Did the team create some-thing that has not already been created? Is it unique?	Did the team use APIs available to create a solution?	What is the wow factor? Would others be impressed by what was built? How easy is the solution to use?	Is the project technically impressive / complex? Is it remarkable that a team created this solution in the time allowed?	Was the presentation or demo well put together? Did the team seem prepared? How well did they explain the problem and solution? (only judge on content, not video quality)

INTEROPATHON | 2020

MiHIN MICHIGAN HEALTH INFORMATION NETWORK SHARED SERVICES
INTEROPERABILITY INSTITUTE